

ELECTRONIC

FANTASY

AUTHORIZED ATARI* SERVICE CENTER

GTIA's, ROM C's
REV "B" OS UPGRADE
DISK DRIVE REPAIR
400 or 800 PROBLEMS?

THEN BRING THEM TO US.
FAST TURN AROUND AND ALL WORK

100% GUARANTEED.

VERY REASONABLE RATES,
PLUS A FULL LINE OF ATARI PRODUCTS
AND SOFTWARE.

ITS THE ONLY COMPUTER WE SELL.
NOW WE OFFER THE BAY AREA ATARI USER
GROUP MEMBERS A 10% DISCOUNT ON OUR
REGULAR RATES ON REPAIRS AND SOFTWARE.

LIMITED TIME

THREE LOCATIONS IN
CUPERTINO, SAN MATEO
AND SOUTH SAN FRANCISCO

ELECTRONIC FANTASY
2180 VALLCO
FASHION PARK
CUPERTINO, CA.
408 255-9451

ELECTRONIC FANTASY
2180 SAN MATEO
FASHION ISL.
SAN MATEO, CA.
415 589-9800

ELECTRONIC FANTASY
COMPUTER SUPPORT
52 S. LINDEN #1
S. SAN FRANCISCO, CA.
415 589-9800

**BAY AREA ATARI USERS GROUP
NEWSLETTER**

is published by the Bay Area ATARI Users Group, an informal association of individuals. This group is not affiliated with Atari, Inc. or any other commercial organization. Any logos, trademarks, or company names are used either with permission or as an informal way of referring to a product or organization.

CORRESPONDENCE ADDRESS:

Bay Area ATARI Users Group
4029 Payne Ave.
San Jose, Calif., 95117
Phone 408-244-7181

All articles are written and donated by members. Opinions expressed in this publication are those of the individual author and do not necessarily represent or reflect the opinions of the Bay Area Atari Users Group as a whole or any other commercial or non-commercial organization.

Deadline

Newsletter submissions must be in by the third Tuesday of each month. If the material is not of a time dated nature, it may not be published in the newsletter of the following month. The editor reserves the right to decide if articles and programs shall or shall not be published in a given month.

Please submit articles camera ready, typed or printed in 3 1/2 inch columns. Mail to "EDITOR" at the correspondence address.

Subscriptions (which include Group membership) are \$12 per year and entitle you to 12 newsletters. Single newsletters are \$1.

BAY AREA
ATARI*
USERS GROUP
4029 Payne Avenue
San Jose, Calif.
95117

BAY AREA ATARI* USERS GROUP MAY MEETINGS

The next regularly scheduled meeting of the Bay Area Atari Users Group will be Monday May 3rd at 7:00 PM. The meeting will be at our usual location: Dysan, Inc.; 5201 Patrick Henry Way, Santa Clara.

Our featured speaker will be Steve Gerber, manager of the ATARI Regional Software Acquisition Center in Sunnyvale. He will talk about the Center and the services they provide for software authors.

Additionally, Robin Ziegler will give us another demonstration on speech syNthesis using FORTH and POKEY, the built-in sound generator chip in the computer.

Our midmonth meeting this month will at be 7:00 PM, Tuesday, May 18th at Computer Capers in the Old Mill Shopping Center in Mountain View. If you still haven't seen this store with computers to rent, you should make it a point to come for sure.

CLYDE

The major news this month is the acquisition of fig-FORTH 1.4. This is a truly excellent version of FORTH for the ATARI*. It is available, free, to anyone who wants it. It includes a disk speed checker, a decompiler which even decompiles primitives, a screen editor, a player/missile module which puts a player on GR. 7 screen and lets it be controlled by joysticks, a stack display at the top of the screen, and novice tips on screen 40. For anyone with the slightest interest in FORTH for the ATARI* this is a package that can't be beaten even for a hundred bucks and all it costs is your disk, or \$3 if you want to just buy a club disk. See Dave Flory at a meeting or call for an appointment to get your copy.

BAY AREA ATARI USER GROUP OFFICERS

CHAIRPERSON: DAVE FLORY
4029 PAYNE AVE.
SAN JOSE, CA., 95117
408-244-7181

SPEAKERS: CLYDE SPENCER
1714 B MARSHALL CT.
LOS ALTOS, CA., 94022
415-965-9355

NEWSLETTER: SOMEONE NEEDED

ADVERTISING SALES: SOMEONE NEEDED

TREASURER: VICTOR ROGERS
17595 MANZANITA DR.
MORGAN HILL, CA., 95037
408-779-0185

DOCUMENTS: BOB LINDEMAN

408-984-2228

SECRETARY: ROBERT KAWARATANI
2460 W. BAYSHORE #8
PALO ALTO, CA., 94303
415-856-3579

PROGRAMS:

DISKS: CHIEF LIBRARIAN
GEORGE AMY
415-489-4732

TAPES: ROBERT HINDS
4944 NEW WORLD DR.
SAN JOSE, CA., 95136
408-226-4513

PROBLEM HOTLINE

EVENING	CONTACT	PHONE NUMBER	SPECIALTY
MONDAY	Robin Ziegler	408-438-6879	FORTH & Assembly
TUESDAY	Chris McAfee	408-258-8442	Hardware
WEDNESDAY	John Crane	408-268-7317	Basic & FORTH
THURSDAY	Paul Conrad	408-226-7676	Basic
FRIDAY	Skip Inskeep	408-251-5517	FORTH
SATURDAY	Bob Burkhalter	408-856-1893	Disc I/O
SUNDAY	Hans Hansen	415-490-0175	DOS I & II

Please call only between 7 P.M. and 9 P.M. on the night the individual is listed as being on call. If you can't get an answer try one

Review of Ken Skier's
"Beyond Games: Systems Software
for your 6502 Personal Computer"

by Mabry Tyson

Ken Skier's "Beyond Games: Systems Software for Your 6502 Personal Computer" was reviewed. The book de-scribes a "visible monitor" which displays a memory address, its contents, and the contents of the registers. You can alter the contents of any memory location, JSR to a location with the registers set to particular values, shift bytes around in memory, or call a very simple text editor.

The review of this book in the February issue of Byte magazine was fairly accurate but it did not notice that the code for Ataris does not work! Here are some of my thoughts on the book and the corrections necessary for Ataris.

I am experienced with assemblers on mainframes but I bought the book in order to learn how to program for the 6502. I found it to be a reasonable introduction but I needed another book for a better introduction to the machine instructions. (As you can see in the corrections below he makes at least one erroneous statement.)

I went through all the code and typed it in (learning as I did this) but when I got done, the code didn't work. In fact, although Mr. Skier claims that his program runs on the Atari, he published code that had NEVER been tried. If he had loaded his code into an Atari he would have found that typing any character would have caused that character to be read and reread infinitely.

There were three major bugs in the Atari package all of which were not too difficult to find (two were found during entry). A much harder bug to track down was not in the separate Atari module but apparently doesn't cause much problems on the other 6502's. It totally disrupted the screen on the Atari.

I was also slightly displeased with his style of coding. Part of the problem was that he was coding for a resident assembler which are usually fairly primitive. No mention was made of macros that greatly ease code development on more powerful assemblers. He divided his program into separate modules (which is a good practice) but reused label names in each of the modules. This cause problems when I combined all the modules so I could relocate the entire program more easily.

Despite these problems I like the basic design of his program. I am continuing to use it and have added a breakpoint routine and am adding a single stepper to facilitate debugging my assembly code.

Typos and Errors

In Chapter 3, Loops and Subroutines:

Page 23, footnote and Table 3.1

"If you wish to test the status of the carry flag after a compare, you must set it (using the instruction SEC) before the compare." is not true.

In Appendix A4, 6502 Instruction Set - Opcode List:

Page 175, instruction 1E.

Should be "ASL - Absolute,X" rather than "Future Expansion".

Page 176, instruction 3E.

Should be included. "3E -- REL - Absolute,X".

Page 176, instruction 56.

Should be "LSR - Zero Page, X".

Page 178, instruction E6.

Should be "INC - Zero Page".

In Clear Portion of Screen:

Page 216, listing line 1850, code address 1129.

Should be "BEQ CLRROW" rather than "BPL CLRROW".

In table of mnemonic codes:

Page 308, listing line 1910, code address 1C9A.

Should be \$A6 rather than \$A3.

Page 309, listing line 1950, code address 1CBA.

Should be \$A0 rather than \$9E.

Page 309, listing line 2000, code address 1CE1.

Should be \$88 rather than \$85.

In Table of Addressing Mode Codes:

Page 312, listing line 2420, code address 1D49.

Should be 4 rather than 12.

("BYTE 18,4,2,0,12,12,12,0")

Page 314, listing line 2570, code address 1DB8.

Should be 18 rather than 20.

("BYTE 18,16,18,0,14,14,15,0")

In Text Editor: Update Subroutine:

Page 344, listing line 3320, code address 1F56.

Should be "BNE *+4" rather than "BNE *+6".

In Screen Parameters for the Atari (32K):

Page 376, listing line 760, code address 1000.

Should be "HOME .WORD \$7C40" rather than "HOME .WORD \$7C42".

In Get an Ascii Character from the Keyboard:

Page 378, listing line 2280, code address 102F.

This TAY should be followed by

LDA	##FF	Must clear the key
STA	\$02FC	after reading

Also on page 209.

In Keyboard Definition Table for the Atari:

Page 383, listing line 4410, string beginning at address 0F4D.

Should be 'I|V' rather than 'I-= '.

Page 383, listing line 4420, string beginning at address 0F55.

Should be 'BXZ\$' rather than 'BXZ4'.

The user interface to any program is very important. We've all used or written programs that could be halted by the improper input of information. This is very annoying and frustrating to both the user and the programmer. Consequently, the programmer must make a trade off--a simple program, that requires the user to be careful about the keys that are pressed, or a complex program which is bomb proof. I have found (as have many other programmers) that the use of menus solves both problems nicely. Menus provide for fool proof user interfacing with a minimal programming overhead.

A menu is just what it sounds like--a list of options. This list restricts the user to single character inputs. If an invalid input is received, it is ignored by the program. One nice feature, that is sometimes used, is a moveable cursor. This cursor is controlled by either the keyboard or an optional joystick. This has the advantage that the cursor movement is restricted to only valid options and it is not possible for the user to inadvertently do anything to halt the program.

I have written a general subroutine that can accomodate many different menu lengths and menu formats. This subroutine uses the optional joystick attachment to the Atari. If you do not have a joystick, you can easily adapt the subroutine to use the keyboard instead.

The first thing you should notice is that several parameters need to be setup by the calling program. These define the length, format, and location of the menu. In the example, there are 48 items in the list, there are two columns of 24, with 15 characters separating the columns. You should also note that the items in the menu are written before the actual selection routine is called. By writing the menu as I have done, it can be a simple subroutine call using data statements for the items, making it very easy to have several menus in a program.

The subroutine returns the value of the selected item in the variable 'T'. You should also note that I have used sounds in this routine. I'm a firm believer in the effectiveness of audio feedback in a program, and write all of my programs appropriately.

HAPPY PROGRAMMING

Don Lang

```

2000 ? "3"
2005 REM ***** A IS THE X-POS OF THE FIRST MENU ITEM
2010 REM ***** B IS THE Y-POS OF THE FIRST MENU ITEM
2015 REM ***** M IS THE NO. OF SPACES BETWEEN THE TWO COLUMNS
2020 REM ***** MAX IS THE NUMBER OF ITEMS
2025 REM ***** T IS THE RETURNED SELECTED ITEM NUMBER
2027 REM ***** NOW SETUP THE PARAMETERS OF THE MENU
2030 A=5:B=0:M=15:MAX=48:MAX2=24
2031 REM ***** CALL SUBROUTINE TO LIST ITEMS TO SCREEN
2032 RESTORE 3000:GOSUB 2530
2035 T=1:X=A:Y=B
2037 REM ***** CALL SUBROUTINE
2040 GOSUB 2050
2045 GOTO 2000
2047 REM ***** BLINK CURSOR AND LOCATE IT AT THE FIRST LOCATION
2050 POKE 752,1:POSITION X,Y:POSITION X-1,Y:POKE 752,0:?" ";
2052 REM ***** THE NEXT LINES MOVE CURSOR
2055 IF STICK(0)=13 THEN T=T+1:SOUND 3,T*4,10,8:FOR I=1 TO 3:NEXT I:SOUND 3,0,0,
0
2060 IF STICK(0)=14 THEN T=T-1:SOUND 2,T*4,10,8:FOR I=1 TO 3:NEXT I:SOUND 2,0,0,
0
2065 IF T>MAX THEN T=MAX
2070 IF T>MAX2 THEN X=A+M:Y=B+T-(MAX2+1)
2075 IF T<1 THEN T=1
2080 IF T<=MAX2 THEN X=A:Y=B+T-1
2082 REM ***** TEST TO SEE IF CURRENT ITEM IS SELECTED
2085 IF STRIG(0)=0 THEN FOR I=1 TO 10:SOUND 1,33+I*10,10,8:NEXT I:SOUND 1,0,0,0:
RETURN
2090 GOTO 2050
2530 FOR I=0 TO MAX2-1:J=A+2:READ ITEM
2540 POSITION J,B+I:?" ITEM";:NEXT I
2545 J=J+M
2550 FOR I=0 TO MAX-MAX2-1:READ ITEM
2560 POSITION J,B+I:?" ITEM";:NEXT I
2570 RETURN
3000 DATA 1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6
3010 DATA 27,28,29,30,1,2,3,4,5,6,7,8,9,40,1,2,3,4,5,6,7,8

```


SOFTWARE REVIEW

Ever since my ten year old daughter saw an arcade car driving program at the Santa Cruz boardwalk, she has asked why such a program isn't available for our computer. The few out for the Atari are quite simple and induce boredom quickly. Match Racer is the best program of this type I have seen on the Atari. Unfortunately, it doesn't compare to the Arcade versions. The program runs in 16k and is all machine language. It starts out with the nicest ragtime music theme I have heard. The screen has lots of color and is fine scrolled vertically for the race track. The roadway has traversable brick sides, causing your vehicle to bounce and chatter. There are grease and water slicks, trees, barricades etc. It is a one or two player game, and the screen displays the odometers. At 5 miles down the road, an automatic increase in speed takes place, which is then increased again at 15 & 45 miles down the road. When your vehicle is destroyed by an obstacle, a crash sound is heard, followed by an annoying and long-sounding siren. If you select the one player option, the game wants to reset itself to two player at the end of the game, causing one to re-select the one player option again. Each driver begins with three cars in single player mode and five cars in two player mode. Each crash removes a car until the driver destroys the final vehicle. Drivers are awarded additional cars after 10 miles. The game is over when all cars are gone. All in all—the best I have seen to date. However, I will be keeping my eye open for more real-looking terrain, cars and action in future programs that allow me to drive.

Robert Lewis

Miscellany

This month I'll look at a grab bag of items. Two new programs and one not so new program which continues to be highly useful. The column ends with a little routine that I translated myself (a first).

Disk Boot Maker from Computer Age Software is designed to make boot disks from boot tapes. I bought it because I am impatient and don't like waiting for my tapes to load and would much rather use the disk drive. In particular, it makes moving programs such as Space Invaders from tape to disk very easy. A useful product for those moments of need.

Weekly Planner by Ronald and Lynn Marcuse from APEX is another worthwhile product. It allows one to store important dates and appointments on a floppy disk. Weekly Planner can printout formatted monthly calendars and daily appointment listings. It is simple to use and can be very useful in putting some order in ones life. Professionals could find a useful supplement to maintaining calendars.

There are probably few members of this club who have not heard of File Manager 800. Switching from the Atari Mailing List program to File Manager for handling the club mailing list has made life much more pleasant when mailing label printing time rolls around. File Manager is simultaneously much easier to use, 20 times faster, and infinitely more efficient than the Atari Mailing list. If the club would only buy a printer equipped with adjustable tractor so I could use sprocketed continuous labels instead of the labels I am using now to print the mailing labels.

For the members of the group interested in geometry, the combination of Pilot and Turtle Geometry is a dynamite combination. The following routine was adapted from a Logo program in Turtle Geometry. By varying the angle, one can draw an interesting variety of polyspirals (try 90, 95, 117, and 120 degrees).

Robert Kawatani

Getting Started in Assembly Language

For your ATARI computer, assembly language is the programming language offering the greatest flexibility, memory efficiency, and speed. It may also be the most difficult language to learn and to use. It requires that you master three subjects:

- 1) Using the ATARI Assembler/Editor (or equivalent)
- 2) Understanding 6502 Assembly Language and Machine Language
- 3) Using The ATARI Computer Hardware and Operating System

Mastering subject #1 is straightforward. All you need is the ATARI Assembler Editor (\$40-\$60) and the manual that comes with it. If you learned ATARI BASIC from Atari's literature, you can learn to use the ATARI Assembler Editor from Atari's manual. It is very well written; however, it will not help you with subjects #2 and #3. (Incidentally, there are other assemblers available. Make sure that any assembler you buy has an editor, assembler, disassembler, and debugger, as Atari's does. Synapse Software's soon-to-be-released assembler is reputed to be superior to Atari's, and will sell at a competitive price.)

Mastering subject #2 requires a good book on 6502 Assembly Language programming. You might try one of the following:

- 6502 Assembly Language Programming, by Lance Leventhal (\$17)
Programming the 6502, by Rodney Zaks (\$16)

I've seen another assembly language book aimed specifically at ATARI computer owners. (I can't remember the title or author.) I wouldn't rate it any better or worse than the other books for use with the ATARI.

I used Lance Levanthal's book when I was learning. It's better for people with some prior assembly language experience, or who buy the companion volume that introduces microprocessors and assembly languages in general.

Even after you've learned 6502 Assembly Language and can use your assembler/editor, you must still learn how to apply it to the ATARI system (subject #3). For example, you may want to manipulate character sets, draw and move players and missiles, produce complex sounds, or perhaps do some fine scrolling. None of the previously mentioned literature shows you how to do these.

You're stuck with buying the ATARI Hardware Reference and Operating System Manuals (\$30 for the set) and De Re Atari (\$20 from APX).

The Hardware Reference and Operating System Manuals list all special memory locations and describe all special features that you might want to use. The manuals are poorly written and difficult to understand, but are very complete and contain information that you can't find elsewhere. The Master Memory Map (\$7 from Santa Cruz Educational Software) is a low-cost substitute for those who don't need all the detail in the ATARI manuals. You may find the Master Memory Map easier to read and understand.

De Re Atari is a book describing the ATARI 400/800 computers. It's the only literature I know of that can actually teach you to use the ATARI's "hidden" features with assembly language programming. The book includes information on Display Lists, Player Missile Graphics, Display List Interrupts, Scrolling, the Operating System, Sound, and a variety of other subjects. It is being published one chapter at a time in Byte magazine. If you are particularly interested in one or two of the subjects listed above, you might check out the appropriate issues of Byte from the library rather than spend \$20 for the book.

When you have mastered all three subjects (#1, #2, and #3), you will best be able to exploit the full capabilities of your machine. You can enhance your programs written in other languages with machine language subroutines, or write entire programs in assembly language.

Gray Chang

MX-80 GRAFTRAX SCREEN DUMP

by RON MILLER

This program dumps the ATARI graphics window generated in graphics mode 8. There are no GOTO's or GOSUB's in this program so it can be renumbered and placed into a subroutine without affecting program logic. This program uses memory locations HEX 0600-0609 for data storage and it uses I/O channel 3. The program has one machine language routine stored inside T\$. Bytes can be read into T\$ via BASIC or they can be keyed in directly from the keyboard. Using the keyboard is easier.

```

10 REM -----
12 REM T$ CONTAINS ASM TRANSPOSE
14 REM
15 DIM T$(23)
16 T$="..T$.goes.here....."
20 REM -----
22 REM OPEN A PRINTER OUTPUT CHANNEL
24 REM
30 OPEN #3,4,0,"P:"
40 REM -----
42 REM SET PRINT SPACES TO 24/216 IN.
44 REM
50 PUT #3,27:PUT #3,51:PUT #3,24
60 REM -----
62 REM SEND 'CR' TO PRINTER
64 REM
70 PUT #3,155
100 REM -----
102 REM LOCATE DISPLAY LIST
104 REM
110 DL=PEEK(561)*256+PEEK(560)
120 REM -----
122 REM LOCATE GRAPHICS SCREEN DATA
124 REM
130 SD=PEEK(DL+5)*256+PEEK(DL+4)
140 REM -----
142 REM BEGIN OUTPUT LINES
144 REM
150 FOR LINES=0 TO 19
152 PUT #3,27:PUT #3,75:PUT #3,64:PUT #3,1
160 REM -----
162 REM BEGIN OUTPUT CHARACTERS
164 REM
170 FOR CHARS=0 TO 39
180 REM -----
182 REM READ 8 SCREEN BYTES
184 REM
190 FOR BYTE=0 TO 7
200 BYTEX=SD+BYTE*40+CHARS+LINES*320
202 POKE 1536+BYTE,PEEK(BYTEX)
210 NEXT BYTE
220 REM -----
222 REM CONVERT BYTES FOR PRINTING
224 REM
230 X=USR(ADR(T$))
240 REM -----
242 REM OUTPUT CONVERTED BYTES
244 REM
250 FOR I=0 TO 7
260 PUT #3,PEEK(1544+I)
270 NEXT I
280 REM -----
282 REM
284 REM
290 NEXT CHARS
300 REM -----
302 REM SEND 'CR' TO PRINTER
304 REM
310 PUT #3,155
320 REM -----
322 REM
324 REM
330 NEXT LINES
340 REM -----
342 REM SEND 3 'CR' TO PRINTER
344 REM
350 PUT #3,155:PUT #3,155:PUT #3,155
400 REM -----
402 REM END OF PROGRAM

```

T\$ contains 23 bytes as follows:

BYTE#	HEX	DEC	KEYBOARD
1	68	104	h
2	A0	160	inverse space
3	00	0	cntl ,
4	A2	162	inverse "
5	07	7	cntl G
6	1E	30	esc cntl +
7	00	0	cntl ,
8	08	8	cntl F
9	6A	106	j
10	CA	202	inverse J
11	E0	224	inverse cntl
12	FF	255	esc cntl >
13	D0	208	inverse P
14	F7	247	inverse w
15	99	153	inverse cntl
16	08	8	cntl H
17	06	6	cntl F
18	C8	200	inverse H
19	C0	192	inverse @
20	08	8	cntl H
21	D0	208	inverse P
22	ED	237	inverse m
23	60	96	cntl .


```

10 REM "PRINT USING" SUBROUTINE
12 REM FOR ATARI COMPUTER
14 REM BY ROBIN LAU
16 REM FEB 26, 1982
18 REM
100 DIM USE$(20), FLAG$(2)
102 REM USE$: USER DEFINED PRINT FORMAT
104 REM FLAG$="+" OR "-"
105 REM
110 USE="00"
112 REM STARTING SUBROUTINE LINENO.
113 REM
120 GOTO 1000
122 REM GOTO DEMO PROGRAM
123 REM
200 USE$="00":FLAG$="+":FIX=1
201 REM USE$: USER DEFINED FORMAT
202 REM FLAG$: USER DEFINED FORMAT
203 REM FIX=1: LEFT JUSTIFIED "$" SIGN
204 REM FIX=0: FLOATING "$" SIGN
205 REM
210 IF TEMP<0 THEN FLAG$="-":TEMP=ABS(TEMP)
212 REM IF NO.<0 THEN PUT "-" BEFORE "$"

213 REM
220 TEMP=INT(TEMP*100+0.5)/100
222 REM ROUNDED TO TWO DECIMAL PLACES
223 REM
230 L1=LEN(USE$):L2=LEN(STR$(INT(TEMP)))

240 IF TEMP>=(10^(L1-5)) THEN FOR I=3 TO
  L1:USE$(I,1)="$":NEXT I:L2=L1-5:GOTO 26
0
242 REM NUMBER TOO LARGE TO FORMAT
243 REM
250 USE$(L1-2-L2,L1)=STR$(TEMP)
260 IF FIX THEN USE$(1,2)=FLAG$:RETURN
270 USE$(L1-4-L2,L1-3-L2)=FLAG$:RETURN
1000 REM DEMO
1010 READ NUMBER:TEMP=NUMBER:GOSUB USE
1020 LPRINT USE$,NUMBER
1030 IF NUMBER=0 THEN END
1040 GOTO 1010
1050 DATA 1234567.891,-1234567.891
1060 DATA 123456.789,-123456.789
1070 DATA 12345.6789,-12345.6789
1080 DATA 123.45678,-123.45678
1090 DATA 12.345678,-12.345678
1100 DATA 1.2,-1.2
2000 DATA 0

```

'PRINT USING' SUBROUTINE SAMPLE RUNS :

```

200 USE$="00":FLAG$="+":FIX=1
+***** 1234567.89
-***** -1234567.89
+$123456.79 123456.789
-$123456.79 -123456.789
+$ 12345.68 12345.6789
-$ 12345.68 -12345.6789
+$ 123.46 123.45678
-$ 123.46 -123.45678
+$ 12.35 12.345678
-$ 12.35 -12.345678
+$ 1.20 1.2
-$ 1.20 -1.2
+$ 0.00 0

```

```

200 USE$="00":FLAG$="+":FIX=0
+***** 1234567.89
-***** -1234567.89
+$123456.79 123456.789
-$123456.79 -123456.789
+$12345.68 12345.6789
-$12345.68 -12345.6789
+$123.46 123.45678
-$123.46 -123.45678
+$12.35 12.345678
-$12.35 -12.345678
+$1.20 1.2
-$1.20 -1.2
+$0.00 0

```

```

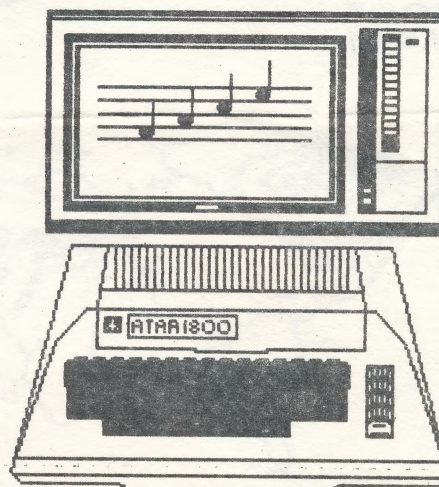
200 USE$=">>>>>>00":FLAG$=" $":FIX=1
***** 1234567.89
-***** -1234567.89
***** 123456.789
-***** -123456.789
$12345.68 12345.6789
-$12345.68 -12345.6789
$>>123.46 123.45678
-$>>123.46 -123.45678
$>>>12.35 12.345678
-$>>>12.35 -12.345678
$>>>>1.20 1.2
-$>>>>1.20 -1.2
$>>>>>0.00 0

```


FROM THE DALLAS USER GROUP "IRATA" NEWSLETTER
ATARI CODES FOR THE EPSON MX-80 PRINTER

The following is a list of frequently used Atari codes for the Epson MX-80 printer. All codes are contained in LPRINT statements, be sure to use quotes. When you see this (CTRL & M) it means to press the CTRL and M keys together. When you see (ESC ESC A X) it means to press ESC twice and then the A key and then whichever keys are needed for "X".

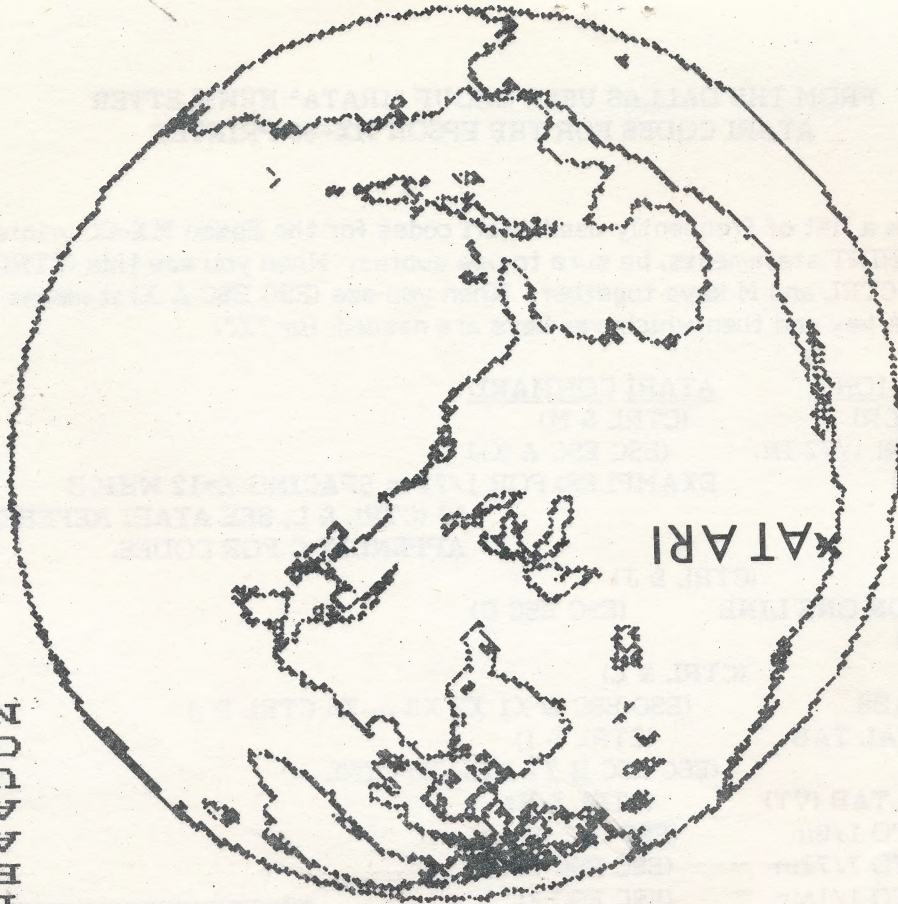
<u>PRINTER FUNCTION</u>	<u>ATARI COMMAND</u>
CARRIAGE RETURN(CR)	(CTRL & M)
SET LINE SPACING IN 1/72 IN.	(ESC ESC A X)J
INCREMENTS (ESC A)	EXAMPLES: FOR 1/72 in SPACING X=12 WHICH IS (CTRL & L, SEE ATARI REFERENCE MANUAL APPENDIX C FOR CODES.
LINE FEED (LF)	(CTRL & J)
SET FORM LENGTH ON ONE LINE	(ESC ESC C)
INCREMENTS (ESC C)	
FORM FEED (FF)	(CTRL & L)
SET HORIZONTAL TABS	(ESC ESC D X1 X2 X3Xn CTRL & ,)
EXECUTE HORIZONTAL TAB	(CTRL & I)
SET VERTICAL TAB	(ESC ESC B X1 X2....Xn CTRL ,)
EXECUTE VERTICAL TAB (VT)	(CTRL & K)
SET LINE SPACING TO 1/8in	(ESC ESC O)
SET LINE SPACING TO 7/72in	(ESC ESC 1)
SET LINE SPACING TO 1/16in	(ESC ESC 2)
SHIFT OUT MEANS DOUBLE WIDTH	(CTRL & N)
CHARACTERS (80)	
CANCEL SHIFT OUT (DC4)	(CTRL & T)
SHIFT IN MEANS CONDENSED	(CTRL & O)
CHARACTERS (SI)	
CANCEL SHIFT IN DC2	(CTRL & R)
EMPHASIZED CHARACTERS	(ESC ESC E)
CANCEL ESC E	(EXC ESC F)
DOUBLE PRINTING	(ESC ESC G)
CANCEL ESC G	(ESC ESC H)
TAB SETTING CODES MUST	(CTRL & ,)
END WITH (NULL)	
RING BELL	(CTRL & G)



```

32600 REM SCREEN PRINTER FOR EPSON
32603 LPRINT
32605 REM GOSUB 32600 IS USED TO PRINT THE CONTENTS OF THE SCREEN ON A PRINTER
32610 OPEN #5,8,0,"P:"
32615 SM=PEEK(106)*256-960
32620 COUNT = 1
32625 FOR X=SM TO SM+959
32626 Z=PEEK(X)
32627 IF Z>128 THEN Z=Z-128
32628 IF Z<96 THEN Z=Z+32
32630 PUT #5,Z
32635 IF COUNT=40 THEN ? #5;CHR$(13);;COUNT = 0
32640 COUNT = COUNT +1
32645 NEXT X
32650 CLOSE #5
32653 LPRINT
32655 RETURN

```

MAY
1982
Bay
Area
ATARI
User
Group
Newsletter



MAY
BAY AREA
ATARI
USERS GROUP
4029 Payne Avenue
San Jose, Calif.
95117



GARY CHANG
2265 DEBORAH DR., #1
SANTA CLARA, CA 95050